NAME RESOLUTION DEVICE AND NAME RESOLUTION METHOD WITH AUTOMATIC NODE INFORMATION UPDATING FUNCTION

5 BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

The present invention relates to a name resolution device and a name resolution method for appropriately
10 updating a name of each node on a network.

DESCRIPTION OF THE RELATED ART

In recent years, the world's largest computer network called Internet has been spreading rapidly, and new
15 computer businesses have been developed by utilizing information or service publicly disclosed by being connected to the Internet, or conversely by providing information or service to external users who make accesses through the Internet. There are also development and
20 evolution of new techniques regarding the Internet utilization.

On the Internet, each computer has an identifier called IP address, and the packet exchanges are carried out based on this IP address. However, the IP address is a
25 numerical value, which is not intuitive for a direct use by the human being so that it lacks the convenience. For this reason, a mechanism for converting the IPv6 address into a character string that is easier to handle for the human being, that is "name", has been used. Currently, the most
30 widely utilized conversion mechanism on the Internet is DNS (Domain Name System).

The DNS is used for a global name resolution on the Internet, so that a name FQDN (Fully Qualified Domain Name) used by the DNS must be uniquely defined on the Internet.
35   On the other hand, due to the spread of the Internet,

-1-

the need for every node to have a globally unique name is
diminishing. Also, in the DNS, because of its name
disclosure property, it is impossible to publicly disclose
a name of some node only to specific members. Consequently,
5 what name does some domain have can be discovered by the
brute force, and a malicious third person can easily
acquire information regarding nodes with what names are
existing. This causes the problem of the privacy leakage.
In addition, in order to utilize the FQDN, there is a need
10 to provide a DNS server for managing that FQDN, and this
gives rise to the problem of the arranging location, the
management, the registration, and the running of this
server.

On the other hand, for a small network such as a home
15 network, there are some propositions for a technique of
defining a name locally on the network and making access to
a node by using that name.

However, these techniques presuppose that the
correspondent specified by the name is located in proximity
20 from a viewpoint of the network, and they cannot be
directly utilized in the case of attempting to make an
access by using the name from an external of that network.
In this case, a dedicated server responsible for the
resolution from the external is required, or a protocol for
25 discovering the server for that purpose is required, so
that the user's convenience is damaged.

In order to avoid these problems, it is possible to
use a method for utilizing a node local database. For
example, in the OS such as Unix[n], this is well known as
30 "/etc/hosts" file.

In the case where some individual wishes the others to
make accesses, it is possible to inform the name to the
others by a method of exchanging data on address and name
directly. However, the static database must be updated
35 manually.

-2-

On the other hand, in the IPv6, it is expected that the renumbering will occur even in a small network such as a home network. Here, the renumbering implies the change of a network prefix to another value on some network.

5   This renumbering definitely occurs in the IPv6 when the user changed a provider, for example, because an address is basically assigned from a provider. At this point, in the case of utilizing the static node local database and information in the database has already been

10 informed to someone, the occurrence of the renumbering must be informed, but there arises a problem of "who should be information of what" from a viewpoint of the privacy leakage. Namely, when an individual who has a node having two names of "foo" and "bar" on the network informs the

15 name of his own node on the network to the others, there can be cases where it is desired to inform only "foo" to some people and inform only "bar" to some other people. In such a case, after the renumbering occurs, there arises a problem of "who should be informed of what".

20

BRIEF SUMMARY OF THE INVENTION

It is therefore an object of the present invention to

25 provide a name resolution device and a name resolution method capable of informing the update information easily without damaging the privacy when the renumbering occurs on the network.

According to one aspect of the present invention there

30 is provided a name resolution device for managing a name of each node connected to a network and an address for identifying each node, comprising: a node information storing unit configured to store a node information containing a name of a node, a network identification

35 information, a prefix indicating a position on the network,

-3-

and an interface identification information of a node, for
each node; a node information collecting unit configured to
collect the node information of other nodes connected to
the network, through the network; and a node information

5   updating unit configured to update the node information
stored in the node information storing unit, according to
the node information of the other nodes collected by the
node information collecting unit, by updating the prefix
stored in the node information storing unit by using the

10 interface identification information contained in the node
information collected by the node information collecting
unit as a key.

　　　According to another aspect of the present invention
there is provided a name resolution method for managing a

15 name of each node connected to a network and an address for
identifying each node, comprising: storing a node
information containing a name of a node, a network
identification information, a prefix indicating a position
on the network, and an interface identification information

20 of a node, for each node; collecting the node information
of other nodes connected to the network, through the
network; and updating the node information stored by the
storing step, according to the node information of the
other nodes collected by the collecting step, by updating

25 the prefix stored by storing step by using the interface
identification information contained in the node
information collected by the collecting step as a key.

　　　According to another aspect of the present invention
there is provided a computer program product for causing a

30 computer to function as a name resolution device for
managing a name of each node connected to a network and an
address for identifying each node, the computer program
product comprising: a first computer program code for
causing the computer to store a node information containing

35 a name of a node, a network identification information, a

-4-

prefix indicating a position on the network, and an
interface identification information of a node, for each
node; a second computer program code for causing the
computer to collect the node information of other nodes
5 connected to the network, through the network; and a third
computer program code for causing the computer to update
the node information stored in the first computer program
code, according to the node information of the other nodes
collected by the second computer program code, by updating
10 the prefix stored in the first computer program code by
using the interface identification information contained in
the node information collected by the second computer
program code as a key.

Other features and advantages of the present invention
15 will become apparent from the following description taken
in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS
20

Fig. 1 is a block diagram showing a schematic
configuration of a name resolution device according to one
embodiment of the present invention.

Fig. 2 is a diagram showing exemplary IPv6 addresses
25 of two nodes connected to a network according to one
embodiment of the present invention.

Fig. 3 is a diagram showing an entry for a name "foo"
shown in Fig. 2, according to one embodiment of the present
invention.

30 Fig. 4 is a diagram showing an entry for a name "bar"
shown in Fig. 2, according to one embodiment of the present
invention.

Fig. 5 is a sequence chart showing an operation
sequence for a first exemplary updating of a database
35 according to one embodiment of the present invention.

-5-

Fig. 6 is a diagram showing exemplary IPv6 addresses of two nodes after the renumbering, according to one embodiment of the present invention.

Fig. 7 is a diagram showing a network arrangement for
5 explaining a second exemplary updating of a database according to one embodiment of the present invention.

Fig. 8 is a sequence chart showing an operation sequence for a second exemplary updating of a database according to one embodiment of the present invention.

10 Fig. 9 is a diagram showing a network arrangement for explaining a modified case of a second exemplary updating of a database according to one embodiment of the present invention.

15

DETAILED DESCRIPTION OF THE INVENTION

Referring now to Fig. 1 to Fig. 9, one embodiment of a name resolution device and a name resolution method
20 according to the present invention will be described in detail.

Fig. 1 shows a schematic configuration of a name resolution device according to one embodiment of the present invention. The name resolution device of Fig. 1 is
25 provided inside each node connected to a network 1. Here, the node is an arbitrary device having a communication function, such as PC, PDA (Personal Digital Assistants), portable telephone, home electronics node, etc.

The name resolution device of Fig. 1 has a name
30 inquiring unit 2, a database 3, a name updating unit 4, a responding unit 5, a network processing unit 6, and a management unit 7.

The name inquiring unit 2 inquires names to the other nodes connected to the network 1, and collects node
35 information of each node. The collected node information is

-6-

stored into the database 3.

In the IPv6 (Internet Protocol version 6) which is a
protocol on the Internet, it is possible to utilize ICMP
(Internet Control Message Protocol) node information query.
5   Note that the storing into the database 3 may be done
manually, but in the following a mechanism for
automatically collecting information will be described.

The name updating unit 4 judges whether the node
information of each node on the network 1 has been changed
10 or not, and if the node information has been changed, the
name updating unit 4 updates the database 3.

The responding unit 5 returns the node information of
the own node via the network 1, in response to an inquiry
from the other nodes. The network processing unit 6
15 controls data transmission and reception via the network 1.

The management unit 7 manages the node information of
each node inquired and collected by the name inquiring unit
2, in terms of name, network ID, network prefix, and
interface ID. Here, the network ID is an identification
20 information of the network 1 itself, the network prefix is
an information indicating a position on the network 1, and
the interface ID is an identification information of an
interface of the node.

For example, suppose that the name inquiring unit 2
25 discovered two nodes connected to some network Nx. These
two nodes have IPv6 addresses as shown in Fig. 2, and one
has a name "foo", the other has a name "bar". In Fig. 2,
"prefixlen 64" indicates that the prefix length of the
network prefix is the upper 64 bits.

30 In this case, the information as shown in Fig. 3 and
Fig. 4 will be stored into the database 3. Namely, as shown
in Fig. 3, the name "foo" has the network ID "3ffe::/64",
the network prefix "3ffe::/64", and the interface ID
"::230:65ff:fela:3525". Also, as shown in Fig. 4, the name
35 "bar" has the network ID "3ffe::/64", the network prefix

-7-

"3ffe::/64", and the interface ID "::203:93ff:fe00:e6ac".

In the following, a set of the name, the network ID, the network prefix, and the interface ID will be referred to as an entry.

5   Each node can assign a unique name to each network ID stored in the database 3, such as "my home", "relative's home", etc.

Even in the case of connecting each node to the external network 1, it is possible to specify the IPv6
10 address of the correspondent by using the name defined by that node (which is not globally unique), by referring to the database 3 possessed by each node.

Each node can inform some entries from the local database 3 possessed by each node, selectively to the
15 others (other nodes). In this way, it is possible to publicly disclose the name only to the others chosen by each node.

It is also possible for the name inquiring unit 2 to regularly and automatically collect the entries of the
20 other nodes on the network 1 to which each node is connected.

The types and the names of the nodes connected to the network 1 will change in time, so that there is a possibility that an entry stored in the database 3 actually
25 does not exist. For this reason, there is a need to update the database 3 regularly.

(First exemplary updating)

The first exemplary updating presupposes the case where the names of the other nodes on the network 1 can be
30 collected by directly moving the node.

Fig. 5 shows an operation sequence for the first exemplary updating of the database 3. First, one node is connected to the network 1, and the name of another node on the network 1 is inquired (step S1). When there is a
35 response from another node (step S2), the management unit 7

-8-

judges whether the prefix has been changed or not (step S3). If it is not changed, the processing is terminated, whereas if it is changed, the database 3 is updated (step S4).

5   At the step S3 described above, the prefix is updated according to the interface ID of each node on the network 1. Namely, in the case where a node that has the same interface ID but a different network prefix exists among the names collected by the name inquiring unit 2, the name

10   updating unit 4 judges that the name has been updated, and updates the name by carrying out an operation to replace the network prefix corresponding to that interface ID in the database 3 by a new prefix.

     For example, suppose that while one node M is away

15   from the network Nx to which two nodes shown in Fig. 2 are connected, the renumbering occurs in this network Nx such that its network prefix has been updated from "3ffe::/64" to "2001::/64". Then, suppose that the node M moves into the network Nx after that, carries out the name collecting,

20   and discovers two nodes shown in Fig. 6.

     The interface IDs of the two nodes shown in Fig. 6 are the same as the interface IDs of the two nodes shown in Fig. 2, so that the name updating unit 4 of the node M judges that the network prefix has been changed from

25   "3ffe::/64" to "2001::/64", and updates the database 3.

     The interface ID described above is not necessarily globally unique, and the first exemplary updating described above is applicable only in the case where the interface ID is in EUI-64 format which is globally unique relatively.

30   It is also possible to carry out the above described updating processing for the database 3 only in the case where the prefix of a default router of the network 1 to which the node belongs has been changed. Alternatively, it is also possible to ask the user whether the updating of

35   the database 3 is permitted or not.

As described, in the first exemplary updating, the updating of the database 3 is carried out by judging that the node information has been updated, for the node information which has the same interface ID and a different
5 network prefix, so that it is possible to automatically update the database 3.

(Second exemplary updating)

The second exemplary updating notifies the name update information only to the nodes whose identifies are known.
10 In the case where the own node is not directly connected to the network 1, such as the case where the information on nodes connected to the network 1 is learned from a third person, for example, even if the names are updated on that network 1, the database 3 cannot be updated
15 unless the update information is learned from a third person. As a method for updating the database 3 in this case, it is possible to use a method in which the database 3 is updated by the third person who has new entries. In this method, the network prefix of all the entries which as
20 the same value as the network ID of the entries before the updating in the database 3 will be changed to the value after the updating. In this way, it is possible to realize the prefix update efficiently, while maintaining the privacy.
25 Fig. 7 shows an exemplary network arrangement for the purpose of explaining the second exemplary updating, and Fig. 8 shows an operation sequence for the second exemplary updating of the database 3. In Fig. 7, it is assumed that the owner of the network Nx is X, and some person A has
30 already learned the node information for two names "foo" and "bar" from the network owner X, as the information regarding the network Nx (step S11). The person A is assumed to be not directly connected to the network Nx.

Then, the renumbering occurs in the network Nx, such
35 that the network prefix is changed (step S12). In this

case, when the network owner X meets the person A, the network owner X must inform that the network prefix has been changed, but suppose that the network owner X has already forgotten what had been informed to the person A
5 before.

At this point, the person A selects one of the network interface IDs that the person A has already learned about the network Nx, and inform its hash value to the network owner X (step S13). Here, the hash value is obtained by
10 inputting the selected interface ID into a hash function.

Note that it is not absolutely necessary to use the hash function, and any one way function capable of realizing the irreversible conversion can be used.

The network owner X searches for an entry that matches
15 the hash value among the entries possessed by the network owner X, and if such an entry is found, the network owner X notifies the content of the entry by removing the corresponding interface ID, to the person A (step S14). Then, the person A updates the own database 3 according to
20 the notified content of the entry (step S15).

By carrying out such a procedure, the network owner X can avoid informing all the entries unnecessarily, even when what had been informed to the person A before. Also, the person A can update the database 3 according to the
25 change of the names on the network 1, even if the person A is not connected to the network 1.

The second exemplary updating described above can also be utilized for the updating by being told by the other while protecting the privacy. For example, as shown in Fig.
30 9, suppose that the person B has already learned only the information regarding the name "foo" from the network owner X, as the information regarding the network Nx. Namely, this is the case where the network owner X does not wish to inform the information regarding the name "bar" to the
35 person B, for example.

It is assumed that the fact that the network owner X
has made the change of the names is not a secret. When the
person A meets the person B, the person A learns the hash
value of the interface ID already possessed by the person
5   B. If this hash value matches with the interface ID
possessed by the person A, the person A informs only the
corresponding entry by removing the interface ID, to the
person B.

In this way, the person B can update the information
10 that the person B has already possessed, but cannot acquire
any new entry. Namely, the person B cannot obtain
information other than the information already known by the
person B. Similarly, the person A cannot obtain information
of a new node on the network Nx from the person B.

15   In this way, the second exemplary updating notifies
the name update information to the correspondent only when
the hash value matches, so that the privacy can be
protected and the security can be improved.

(Third exemplary updating)
20   The third exemplary updating registers the network
prefix in the DNS, and indicates the value of the network
prefix by the FQDN.

In the third exemplary updating, the FQDN is stored as
the network prefix in the database 3 of each node. The
25 conversion between the network prefix and the FQDN is
carried out by the DNS. The DNS is connected to the network
1 separately from each node.

In the case where the network owner X has some FQDN
regarding the own network 1, namely in the case where the
30 network owner X has a domain for the own network 1, the
value of the network prefix in the entry can be given by
the FQDN indicating the prefix, rather than the actual
prefix.

When the need for the address resolution arises, the
35 network prefix corresponding to the FQDN stored in the

-12-

database 3 is inquired to the DNS. The responding unit 5
makes a response by generating the IPv6 address by using
the network prefix returned from the DNS.

    The database 3 of each node stores the value
5 corresponding to the FQDN instead of the network prefix, so
that at a time of the renumbering, it suffices to update
the value of the FQDN. Each node checks the value
corresponding to this FQDN regularly, and when there is a
request for the IPv6 address of some node from another
10 node, the response is made by generating the IPv6 address
from the network prefix corresponding to the latest FQDN
and the interface ID. Namely, it is tedious to inquire the
network prefix corresponding to the FQDN to the DNS every
time there is a request for IPv6 address, so that the
15 network prefix inquired to the DNS in the past is cached,
and when a request for IPv6 address is made from another
node, the network prefix is read out from this cache and
transmitted to another node. In this way, it is possible to
make a response to the address request faster. Of course,
20 it is also possible to check the DNS every time.

    For example, suppose that a domain called Nx.org
exists, that is, the DNS manages information of
"network.Nx.org is 3ffe::" and information of "3ffe:: is
Nx.org" for a reverse look-up.

25   When some node stores the node information into its
own database 3, the reverse look-up of the network prefix
is carried out, and if the FQDN is set in corresponding to
that, the normal look-up of that FQDN is carried out to
confirm that the value coincides, and the FQDN
30 ("network.Nx.org" in this example) is stored into its own
database 3 as the network prefix. The FQDN stored in the
database 3 can be updated by inquiring to the network 1
appropriately.

    In this way, the third exemplary updating stores the
35 FQDN instead of the network prefix in the database 3 of

-13-

each node, so that the node information becomes easier to comprehend compared with the case of storing the network prefix which is a randomized character string.

In the embodiment described above, the exemplary case
5 where each node has the IPv6 address has been described, but the address is not necessarily limited to this and it is also possible to use the IPv4, for example.

As described above, according to the present invention, even when the node information on the network is
10 changed, each node can learn the occurrence of the change by a simple procedure. Also, it is possible to transmit the node information to the other nodes on the network, while maintaining the privacy.

15   It is to be noted that the above described embodiments according to the present invention may be conveniently implemented using a conventional general purpose digital computer programmed according to the teachings of the present specification, as will be apparent to those skilled
20 in the computer art. Appropriate software coding can readily be prepared by skilled programmers based on the teachings of the present disclosure, as will be apparent to those skilled in the software art.

In particular, the name resolution device of the above
25 described embodiments can be conveniently implemented in a form of a software package.

Such a software package can be a computer program product which employs a storage medium including stored computer code which is used to program a computer to
30 perform the disclosed function and process of the present invention. The storage medium may include, but is not limited to, any type of conventional floppy disks, optical disks, CD-ROMs, magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, magnetic or optical cards, or any other suitable
35 media for storing electronic instructions.

-14-

It is also to be noted that, besides those already
mentioned above, many modifications and variations of the
above embodiments may be made without departing from the
novel and advantageous features of the present invention.
5   Accordingly, all such modifications and variations are
intended to be included within the scope of the appended
claims.

10

15

20

25

30

35